Graphical user interface

Description automatically generated

Python Flask Web App Project Plan

\*Creating Project folders.

1-Creat Project folder Python project web App include all below.

\*Setting Flask configuration and Create Project Folder.

1-Create Project Folder.

2-Creat a Virtual environment.

3-Install Flask to Virtual environments.

4-Setup environment variables.

5-Install the Flask-WTF extension.

6-creat requirements folder for library need to be installed such as.

1-Flask

2-Flask-Login

3-Flask-Security==3.0.0

4-Flask-WTF

6-python-dotenv

\*Create development server.

1-Create a config.py module.

2-Create a routes.py for all routes patterns.

3-Creat \_\_init\_\_.pay file in the project package.

4-Creat main.py to run all the program.

5-Creat forms.py to validate login enters.

6-Creat routes.py includes all routes.

\*Create application folder contains.

1-static folder for CSS for styling.

1-images.css to store image for main page in application.

2-img.css to store image for home page in application.

2-java scripts for validation the user inputs(script.js).

\*Create templates folder contains.

1-Will have base html page to using for inheriting among all pages.

2-Html page for main page(login.html).

3-Html page for navigation(nav.html).

4-Html page for footer(footer.html).

5-Html page for register(register.html).

6-Html page for home (home.html).

7-add more html pages as needed for the project.

\*Create login page.

1-Create the template for home page(index.html).

2-Importe the render\_template function.

3-Using Jinga2 template expression to set up templates.

4-add bootstrap link.

5-add block content from other pages.

\*Navigation links and routes.

1-Create navigation menu.

2-Using url\_for function to resolve links.

3-Using decorator routes to bind a function to one or more URL patterns.

\*Create CSV file.

#Create CSV file to store images and recipes.

\*in routes.py import CSV, render\_template, redirect, SQLite, import a another as needed.

\*Config upload path for csv.

\*Create route to read recipes.csv file and display in home page.

\*Create route for registration and connect to SQLite db.

\*Create route for login.

\*Create route to add new recipes and add to recipes.csv.

\*Create route to delete recipe from recipes uploaded.

\*Terminal order will be use.

1-py -3 -m venv venv to set environment.

2-venv\Scripts\activate to activate the venv.

3-$env:FLASK\_ENV = "development" to set development environment.

4- pip install -> all libraries project need.

5-pip freeze > reqirements.txt to create requrments.txt.

6-pip install -r requirements.txt to install all libraries.

7-pip list to check all installed libraries.

8-$env:FLASK\_APP = "main" .

9-flask run to run the project in server.